IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	David E. Lowell	§	Art Unit:	2195
Serial No.:	10/676,922	§		
		§	Examiner:	Eric Charles Wai
Filed:	October 1, 2003	§		
For:	Runtime Virtualization and Devirtualization of I/O Devices by a Virtual Machine Monitor	§	Atty. Dkt. No.:	200309154-1 (HPC.0518US)

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF PURSUANT TO 37 C.F.R § 41.37

Sir:

The final rejection of claims 1-66 is hereby appealed.

I. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF THE CLAIMS

Claims 1-66 have been finally rejected and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

No amendment after final rejection has been submitted.

Date of Deposit: May 27, 2008

I hereby certify that this correspondence is being electronically transmitted to the U.S. Patent Office on the date indicated above.

Ginger Yount

Ginger Yount

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

Independent claim 1 recites in a computer including an I/O device, a method comprising using a virtual machine monitor (Fig. 1:114) to commence virtualization of the I/O device at runtime (Spec., p. 4, ¶ [0020]; p. 5, ¶ [0023]; Fig. 2).

Independent claim 17 recites in a computer including hardware, a virtual machine monitor (Fig. 1:114) running on the hardware (Fig. 1:110), an operating system running on the virtual machine monitor, the hardware including an I/O device, the I/O device already virtualized by the virtual machine monitor, a method comprising devirtualizing the I/O device at runtime (Spec., p. 4, ¶ [0020]; p. 13, ¶ [0045]; p. 14, ¶ [0049]; Fig. 6).

Independent claim 29 recites a computer comprising:

hardware (Fig. 1:110) including an I/O device; and

computer memory (Fig. 1:115) encoded with a virtual machine (Fig. 1:114) for running on the hardware and commencing virtualization of the I/O device at runtime (Spec., p. 4, ¶ [0020]; p. 5, ¶ [0023]; Fig. 2).

Independent claim 38 recites a computer comprising:

hardware (Fig. 1:110) including an I/O device; and

computer memory (Fig. 1:115) encoded with a virtual machine monitor (Fig. 1:114) for devirtualizing the I/O device at runtime (Spec., p. 4, ¶ [0020]; p. 13, ¶ [0045]; p. 14, ¶ [0049]; Fig. 6).

Independent claim 47 recites an article for a computer including an I/O device, the article comprising software for commencing virtualization of the I/O device at runtime (Spec., p. 4, ¶ [0020]; p. 5, ¶ [0023]; Fig. 2).

Independent claim 56 recites an article for a computer including an I/O device, the article comprising computer-readable memory encoded with software for causing the computer to devirtualize the I/O device at runtime (Spec., p. 4, ¶ [0020]; p. 13, ¶ [0045]; p. 14, ¶ [0049]; Fig. 6).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 17-28, 38-46, and 56-66 Rejected Under 35 U.S.C. § 103(a) as Unpatentable Over U.S. Patent No. 6,075,938 (Bugnion) in View of U.S. Patent Application Publication No. 2002/0083110 (Kozuch).

B. Claims 1-8, 10, 12-14, 16, 29-34, 36, 37, 47-51, and 53-55 Rejected Under 35 U.S.C. § 102(b) as Anticipated by Bugnion.

C. Claims 9, 11, 15, 35, and 52 Rejected Under 35 U.S.C. § 103(a) as Unpatentable Over Bugnion alone.

VII. ARGUMENT

The claims do not stand or fall together. Instead, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-headings as required by 37 C.F.R. § 41.37(c)(1)(vii).

A. Claims 17-28, 38-46, and 56-66 Rejected Under 35 U.S.C. § 103(a) as Unpatentable Over U.S. Patent No. 6,075,938 (Bugnion) in View of U.S. Patent Application Publication No. 2002/0083110 (Kozuch).

1. Claims 17, 18, 20-22, 24, 25, 28.

Independent claim 17 was erroneously rejected as being obvious over the purported combination Bugnion and Kozuch.

To make a determination under 35 U.S.C. § 103, several basic factual inquiries must be performed, including determining the scope and content of the prior art, and ascertaining the

differences between the prior art and the claims at issue. *Graham v. John Deere Co.*, 383 U.S. 1, 17, 148 U.S.P.Q. 459 (1965). Moreover, as the U.S. Supreme Court held, it is important to identify a reason that would have prompted a person of ordinary skill in the art to combine reference teachings in the manner that the claimed invention does. *KSR International Co. v. Teleflex, Inc.*, 127 S. Ct. 1727, 1741, 82 U.S.P.Q.2d 1385 (2007).

Here, even if Bugnion and Kozuch could be properly combined, the hypothetical combination would not have led to the claimed invention.

Claim 17 recites:

> In a computer including hardware, a virtual machine monitor running on the hardware, an operating system running on the virtual machine monitor, the hardware including an I/O device, the I/O device already virtualized by the virtual machine monitor, a method comprising devirtualizing the I/O device at runtime.

The Examiner conceded that Bugnion fails to teach "devirtualizing the I/O device at runtime." 2/6/2008 Office Action at 7. Instead, the Examiner cited Kozuch as purportedly disclosing the claim feature not present in Bugnion. *Id.*

Specifically, the Examiner cited ¶ [0027], lines 7-11, of Kozuch. Paragraph [0027] of Kozuch refers to a method of identifying a decrease in power available to a host platform. The cited paragraph also notes that a determination is made as to whether any of the virtual machines is quiescent. If so, a VMM saves the current state information of the quiescent virtual machine and stops the virtual machine. As a result of stopping the virtual machine, resources allocated to the virtual machine are freed, so that the VMM can reduce power consumption of the one or more devices which were partially or fully allocated to the stopped virtual machine.

Stopping a quiescent virtual machine to free resources allocated to the virtual machine, as taught by Kozuch, has nothing to do with "devirtualizing the I/O device at runtime," as recited in claim 17. All that has occurred in Kozuch is that a virtual machine (which is a software process)

has been stopped. In fact, as depicted in Fig. 1 of Kozuch, there are multiple virtual machines running in the system. Stopping one virtual machine does not change the fact that the VMM 104 of Fig. 1 of Kozuch is still virtualizing the I/O devices (in host platform 108 of Fig. 1) of the system for other virtual machines in the system. Therefore, it is clear that stopping a virtual machine in Kozuch has nothing to do with devirtualizing an I/O device at runtime, as recited in claim 17.

Therefore, even if Bugnion and Kozuch could be hypothetically combined, their hypothetical combination would not have led to the claimed invention. This is a first reason that the obviousness rejection of claim 17 over Bugnion and Kozuch is defective.

Moreover, no reason existed that would have prompted a person of ordinary skill in the art to combine the teachings of Bugnion and Kozuch. As conceded by the Examiner, the concept of devirtualizing an I/O device at runtime is not disclosed in Bugnion. However, contrary to the assertion by the Examiner, Kozuch would also not have provided any teaching or hint to a person of ordinary skill in the art of devirtualizing an I/O device at runtime. Rather, Kozuch teaches stopping a quiescent virtual machine; however, Kozuch's system would continue to virtualize I/O devices using the VMM even after the quiescent virtual machine has been stopped. Therefore, a person of ordinary skill in the art looking to the teachings of Bugnion and Kozuch would not have been prompted to modify the teachings of Bugnion with Kozuch to perform devirtualizing of an I/O device at runtime.

In view of the foregoing, it is clear that the obviousness rejection of claim 17 and its dependent claims over Bugnion and Kozuch is defective.

Reversal of the final rejection of the above claims is respectfully requested.

2. Claim 19.

Claim 19 depends from claim 17, and is therefore allowable for at least the same reasons as claim 17. Moreover, claim 19 further clarifies what is meant by devirtualization, which includes "stopping I/O device emulation at runtime." Stopping a virtual machine (a software process), as taught by Kozuch, is not the same, and provides no hint, of stopping I/O device emulation. Therefore, claim 19 is further allowable for the foregoing reason.

Reversal of the final rejection of the above claim is respectfully requested.

3. Claim 23.

Claim 23 depends indirectly from claim 17, and is therefore allowable for at least the same reasons as claim 17.

With respect to claim 23, the Examiner conceded that both Bugnion and Kozuch "do not teach that the VMM logs I/O accesses by the operating system to the I/O device during devirtualization, and replays the log to the device after devirtualization, whereby the I/O accesses by the operating system are deferred during the devirtualization of the I/O device." 2/6/2008 Office Action at 9.

However, the Examiner argued that Bugnion teaches "that requests must be intercepted and remapped when virtualization of the I/O device occurs (col. 2 lines 57-59)," and that therefore it would have been obvious that "some sort or [sic] logging and playback must occur when the device is devirtualized." *Id.* The cited passage in column 2, lines 57-59, of Bugnion refers to virtualizing access of I/O devices, in which requests are intercepted by the VMM and remapped. There is absolutely no hint given here of a VMM logging I/O accesses by the operating system to the I/O device **during devirtualization**, and **replaying** the log to the device **after devirtualization**.

Therefore, the Examiner's assertion that this feature of claim 23 would be obvious is not supported by any evidence of record.

The obviousness rejection of claim 23 is further defective for the foregoing reason.

Reversal of the final rejection of the above claim is respectfully requested.

4. Claim 26.

Claim 26 depends from claim 17, and is therefore allowable for at least the same reasons as claim 17. Moreover, with respect to claim 26, the Examiner conceded that both Bugnion and Kozuch "do not teach configuring the hardware so that accesses by the operating system to the I/O device no longer trap to the virtual machine monitor." 2/6/2008 Office Action at 10. Instead, the Examiner cited column 11, lines 31-34, of Bugnion as purportedly providing a hint of the claimed feature. The cited column 11 passage of Bugnion refers to a trap such as a page fault, system call, or bus error occurring, and the processor trapping to the monitor that emulates the effect of the trap on the currently scheduled virtual processor. This teaching of Bugnion is actually contrary to the subject matter of claim 26, which refers to configuring the hardware so that accesses by the operating system to the I/O device **no** longer trap to the virtual machine monitor. The cited column 11 passage of Bugnion actually teaches that trapping to the monitor has to occur, which is the opposite of the subject matter of claim 26.

Therefore, the obviousness rejection of claim 26 is further defective for the foregoing reason.

Reversal of the final rejection of the above claim is respectfully requested.

5. Claim 27.

Claim 27 depends from claim 17, and is therefore allowable for at least the same reasons as claim 17. Moreover, with respect to claim 27, the Examiner conceded that both Bugnion and Kozuch "do not teach that wherein the I/O device has multiple modes of operations; wherein the virtual machine monitor determines the mode of the I/O device prior to commencing devirtualization; and wherein the virtual machine monitor restores the determined mode of the operation after devirtualization." 2/6/2008 Office Action at 10-11.

Instead, the Examiner cited column 6, lines 17-20, of Bugnion as providing a hint of the above subject matter. The cited column 6 passage refers to operating systems executing on a plurality of virtual machines and transparently sharing the plurality of computer processors and physical resources through the VMM. There is absolutely no hint given here by Bugnion, or anywhere in the cited references, of the VMM determining a mode of the I/O device prior to commencing **devirtualization**, and then restoring the determined mode of operation after **devirtualization**.

The obviousness rejection of claim 27 is therefore further defective for the foregoing reason.

Reversal of the final rejection of the above claim is respectfully requested.

6. Claims 38-41, 43, 46, 56, 58-61, 65, 66.

Independent claim 38 recites hardware including an I/O device, and a computer memory encoded with a virtual machine monitor for devirtualizing the I/O device at runtime.

Independent claim 56 recites an article for a computer including an I/O device, the article comprising computer-readable memory encoded with software for causing the computer to devirtualize the I/O device at runtime.

As explained above with respect to claim 17, the hypothetical combination of Bugnion and Kozuch would not have disclosed or hinted at devirtualizing an I/O device at runtime. Moreover, no reason existed that would have prompted a person of ordinary skill in the art to achieve the claimed invention.

Therefore, claims 38 and 56, and their respective dependent claims, are non-obvious over Bugnion and Kozuch.

Therefore, reversal of the final rejection of the above claims is respectfully requested.

7. Claims 42, 62.

Claims 42 and 62 are dependent from corresponding base claims 38 and 56, and are therefore allowable for at least the same corresponding reasons. Moreover, claims 42 and 62 recite subject matter similar to claim 23 discussed above. Therefore, the obviousness rejection of claims 42 and 62 is further defective for the additional reason presented above with respect to claim 23.

Reversal of the final rejection of the above claim is respectfully requested.

8. Claim 44, 63.

Claims 44 and 63 depend respectively from base claims 38 and 56, and therefore are allowable for at least the same corresponding reasons. Moreover, the obviousness rejection of claims 44 and 63 is also defective for the additional reason presented above with respect to claim 26.

Therefore, reversal of the final rejection of the above claims is respectfully requested.

9. Claims 45, 64.

Claims 45 and 64 depend from corresponding base claims 38 and 56, and are therefore allowable for at least the same reasons as the corresponding claims. Moreover, the obviousness rejections of claims 45 and 64 are further defective for similar reasons as those stated above with respect to claim 27.

Therefore, reversal of the final rejection of the above claims is respectfully requested.

10. Claim 57.

Claim 57 depends from claim 56, and is therefore allowable for at least the same reasons as claim 56. Moreover, the obviousness rejection of claim 57 is further defective for similar reasons as claim 19 stated above.

Therefore, reversal of the final rejection of the above claim is respectfully requested.

B. Claims 1-8, 10, 12-14, 16, 29-34, 36, 37, 47-51, and 53-55 Rejected Under 35 U.S.C. § 102(b) as Anticipated by Bugnion.

1. Claims 1-8, 12-14, 29-33, 36, 37, 47-50, 54, 55.

Independent claim 1 was rejected as being purportedly anticipated by Bugnion. Independent claim 1 recites a method that uses a virtual machine monitor to **commence** virtualization of an I/O device of a computer **at runtime**. Bugnion is simply silent on when virtualization of I/O devices is **commenced**. In fact, Bugnion appears to suggest that the hardware of a machine is virtualized from bootup to shutdown. *See* Bugnion, 4:24-25 (virtual machine monitor virtualizes all the resources of the machine). This is an indication that all resources of the machine are virtualized at all times. Thus, Bugnion cannot satisfy **commencing** virtualization of an I/O device **at runtime**.

Therefore, claim 1 is not anticipated by Bugnion.

Independent claims 29 and 47 are similarly allowable over Bugnion.

Dependent claims are allowable for at least the same reasons as corresponding independent claims.

Reversal of the final rejection of the above claims is respectfully requested.

2. Claims 10, 34, 51, 53.

Claims 10, 34, 51, and 53 depend from corresponding claims 1, 29, and 47, and therefore are allowable for at least the same reasons as the base claims. Moreover, claim 10 recites that the virtual machine monitor can commence the emulation in the middle of an I/O sequence. As purportedly disclosing this feature, the Examiner cited Bugnion, at column 14, line 32-42. The cited passage refers to intercepting all devices accesses from a virtual machine and forwarding them to physical devices for virtualizing access to I/O devices. However, this provides absolutely no teaching whatsoever of "commencing" the emulation in the middle of an I/O sequence. Therefore, claim 10 is further allowable for the foregoing reason.

Dependent claims 34, 51, and 53 are also further allowable for similar reasons.

Reversal of the final rejection of the above claims is respectfully requested.

3. Claim 16.

Claim 16 depends from claim 1, and is therefore allowable for at least the same reasons as claim 1. Moreover, claim 16 recites **devirtualizing** the I/O device at runtime following the runtime virtualization.

It is noted that the Examiner had conceded that Bugnion fails to disclose devirtualizing the I/O device at runtime (in the rejection of claim 17 on page 7 of the 2/6/2008 Office Action). Thus, the rejection of claim 16 is clearly in error.

Reversal of the final rejection of the above claim is respectfully requested.

C. Claims 9, 11, 15, 35, and 52 Rejected Under 35 U.S.C. § 103(a) as Unpatentable Over Bugnion alone.

1. Claims 9, 11, 15, 35, 52.

Claims 9, 11, 15, 35, and 52 depend from corresponding base claims 1, 29, and 47. In view of the defective rejection of base claims over Bugnion, it is respectfully submitted that the obviousness rejection of dependent claims 9, 11, 15, 35, and 52 is also defective.

Reversal of the final rejection of the above claims is respectfully requested.

CONCLUSION

In view of the foregoing, reversal of all final rejections and allowance of all pending claims is respectfully requested.

Respectfully submitted,

Date: May 27, 2008

Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
1616 South Voss Road, Suite 750
Houston, TX 77057-2631
Telephone: (713) 468-8880
Facsimile: (713) 468-8883

VIII. APPENDIX OF APPEALED CLAIMS

The claims on appeal are:

1. In a computer including an I/O device, a method comprising using a virtual machine monitor to commence virtualization of the I/O device at runtime.

2. The method of claim 1, wherein the computer further includes a CPU, wherein the virtual machine monitor is in control of the CPU prior to the runtime virtualization of the I/O device.

3. The method of claim 1, wherein the virtualization is performed transparently to the operating system.

4. The method of claim 1, wherein the I/O device is compatible with the virtualized I/O device.

5. The method of claim 1, wherein the virtualization includes commencing I/O device emulation at runtime.

6. The method of claim 5, further comprising configuring the hardware to trap I/O accesses, and enabling the virtual machine monitor to emulate the I/O device in response to the traps.

7. The method of claim 6, wherein the virtual machine monitor uses memory management to trap the I/O accesses.

8. The method of claim 5, wherein the virtual machine monitor can commence the emulation between I/O sequences.

9. The method of claim 8, wherein the virtual machine monitor commences emulation by intercepting I/O accesses; wherein the virtual machine monitor uses the intercepted I/O accesses to update a state machine, whereby the state machine reflects a state of the I/O device; and wherein the virtual machine monitor examines transitions in the state machine to determine whether the I/O device is in the middle of an I/O sequence.

10. The method of claim 5, wherein the virtual machine monitor can commence the emulation in the middle of an I/O sequence.

11. The method of claim 5, wherein the virtual machine monitor uses a state machine to determine whether the I/O device is in the middle of an I/O sequence, and delays commencing emulation until the state machine indicates that I/O sequence has completed.

12. The method of claim 1, wherein the runtime virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

13. The method of claim 1, wherein I/O device interrupts are directed to an operating system prior to the runtime virtualization of the I/O device; and wherein the I/O device interrupts are directed to the virtual machine monitor during and after the virtualization of the I/O device.

14. The method of claim 1, wherein the virtual machine monitor temporarily pauses an I/O sequence by emulating the I/O device as being busy.

15. The method of claim 1, wherein the I/O device has multiple modes of operations; wherein the virtual machine monitor determines the mode of the I/O device prior to commencing virtualization; and wherein the virtual machine monitor restores the determined mode of the operation after virtualization.

16. The method of claim 1, further comprising devirtualizing the I/O device at runtime following the runtime virtualization.

17. In a computer including hardware, a virtual machine monitor running on the hardware, an operating system running on the virtual machine monitor, the hardware including an I/O device, the I/O device already virtualized by the virtual machine monitor, a method comprising devirtualizing the I/O device at runtime.

18. The method of claim 17, wherein the devirtualization is performed transparently to the operating system.

19. The method of claim 17, wherein the devirtualization includes stopping I/O device emulation at runtime.

20. The method of claim 17, wherein the virtual machine monitor emulates the I/O device prior to devirtualization; and wherein the devirtualization includes allowing the virtual machine monitor to temporarily stop the operating system from commencing a new I/O sequence.

21. The method of claim 20, wherein the virtual machine monitor temporarily stops the operating system by emulating the I/O device as being in a "busy" or "device not ready" state.

22. The method of claim 20, wherein the virtual machine monitor bounds the amount of time the operating system processing is temporarily stopped.

23. The method of claim 20, wherein the virtual machine monitor logs I/O accesses by the operating system to the I/O device during devirtualization, and replays the log to the device after devirtualization, whereby the I/O accesses by the operating system are deferred during the devirtualization of the I/O device.

24. The method of claim 17, wherein the virtual machine monitor waits for I/Os initiated by the virtual machine monitor's driver for the I/O device to complete, and for all expected interrupts from the device to arrive, before ceasing device emulation.

25. The method of claim 17, further comprising re-directing interrupts from interrupt handlers in the-virtual machine monitor to interrupt handlers in the operating system.

26. The method of claim 17, further comprising configuring the hardware so the accesses by the operating system to the I/O device no longer trap to the virtual machine monitor.

27. The method of claim 17, wherein the I/O device has multiple modes of operations; wherein the virtual machine monitor determines the mode of the I/O device prior to commencing devirtualization; and wherein the virtual machine monitor restores the determined mode of the operation after devirtualization.

28. The method of claim 17, wherein the I/O device is virtualized at runtime again after having been devirtualized at runtime.

29. A computer comprising:
hardware including an 10 device; and
computer memory encoded with a virtual machine for running on the hardware and commencing virtualization of the I/O device at runtime.

30. The computer of claim 29, wherein the I/O device is compatible with the virtualized I/O device.

31. The computer of claim 29, wherein the virtualization includes commencing I/O device emulation at runtime.

32. The computer of claim 31, further comprising configuring the hardware to trap I/O accesses, and enabling the virtual machine monitor to emulate the I/O device in response to the traps.

33. The computer of claim 32, wherein the virtual machine monitor uses memory management to trap the I/O accesses.

34. The computer of claim 31, wherein the virtual machine monitor can commence the emulation in the middle of an I/O sequence.

35. The computer of claim 34, wherein the virtual machine monitor uses a state machine to determine whether the I/O device is in the middle of an I/O sequence, and delays commencing emulation until the state machine indicates that I/O sequence has completed.

36. The computer of claim 31, wherein the virtual machine monitor temporarily pauses the I/O sequence by emulating the I/O device as being busy.

37. The computer of claim 29, wherein the runtime virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

38. A computer comprising:

hardware including an I/O device; and

computer memory encoded with a virtual machine monitor for devirtualizing the I/O device at runtime.

39. The computer of claim 38, wherein the virtual machine monitor emulates the I/O device prior to commencing devirtualization; and wherein the virtual machine commences the devirtualization by temporarily stopping an operating system running on the virtual machine monitor from commencing a new I/O sequence.

40. The computer of claim 39, wherein the virtual machine monitor temporarily stops the operating system by emulating the I/O device as being in a "busy" or "device not ready" state.

41. The computer of claim 39, wherein the virtual machine monitor bounds the amount of time the operating system processing is temporarily stopped.

42. The computer of claim 39, wherein the virtual machine monitor logs I/O accesses by an operating system to the I/O device during devirtualization, and replays the log to-the device after devirtualization.

43. The computer of claim 39, wherein the virtual machine monitor waits for I/Os initiated by a virtual machine monitor driver for the I/O device to complete, and for all expected interrupts from the I/O device to arrive, before ceasing device emulation.

44. The computer of claim 38, further comprising configuring the hardware so operating system accesses to the I/O device no longer trap to the virtual machine monitor.

45. The computer of claim 38, wherein the I/O device has multiple modes of operations; wherein the virtual machine monitor determines the mode of the I/O device prior to commencing devirtualization; and wherein the virtual machine monitor restores the determined mode of the operation after the I/O device has been devirtualized.

46. The computer of claim 38, wherein the virtual machine monitor can virtualize the I/O device after having devirtualized the I/O device at runtime.

47. An article for a computer including an I/O device, the article comprising computer-readable memory encoded with software for causing the computer to commence virtualization of the I/O device at runtime.

48. The article of claim 47, wherein the virtualization includes commencing I/O device emulation at runtime.

49. The article of claim 48, wherein the software includes a virtual machine monitor; and wherein the software configures the hardware to trap I/O accesses, and enables the virtual machine monitor to emulate the I/O device in response to the traps.

50. The article of claim 49, wherein the virtual machine monitor uses memory management to trap the I/O accesses.

51. The article of claim 48, wherein the software includes a virtual machine monitor for commencing the emulation in the middle of an I/O sequence.

52. The article of claim 51, wherein the virtual machine monitor includes a state machine for determining whether the I/O device is in the middle of an I/O sequence, the virtual machine monitor delaying the commencement of the emulation until the state machine indicates that the I/O sequence has completed.

53. The article of claim 52, wherein the virtual machine monitor temporarily pauses the I/O sequence by emulating the I/O device as being busy.

54. The article of claim 47, wherein the software includes a virtual machine monitor for emulating I/O device interrupts during the runtime virtualization.

55. The article of claim 47, wherein the software includes a virtual machine monitor for commencing the virtualization of the I/O device at runtime.

56. An article for a computer including an I/O device, the article comprising computer-readable memory encoded with software for causing the computer to devirtualize the I/O device at runtime.

57. The article of claim 56, wherein the devirtualization includes ceasing emulation of the I/O device at runtime.

58. The article of claim 57, wherein the software includes a virtual machine monitor; and wherein the devirtualization includes temporarily stopping an operating system running on the virtual machine monitor from commencing a new I/O sequence.

59. The article of claim 58, wherein the virtual machine monitor temporarily stops the operating system by emulating the I/O device as being in a "busy" or "device not ready" state.

60. The article of claim 58, wherein the virtual machine monitor bounds the amount of time the operating system processing is temporarily stopped.

61. The article of claim 57, wherein the software includes a virtual machine monitor for ceasing the emulation; the virtual machine monitor waiting for I/Os initiated by a virtual machine monitor driver for the I/O device to complete, and for all expected interrupts from the I/O device to arrive, before ceasing device emulation.

62. The article of claim 56, wherein the software includes a virtual machine monitor for logging I/O accesses by an operating system to the I/O device during devirtualization, and replaying the log to the I/O device after devirtualization.

63. The article of claim 56, wherein the software includes a virtual machine monitor, the software configuring the hardware so operating system accesses to the I/O device do not trap to the virtual machine monitor.

64. The article of claim 56, wherein the I/O device has multiple modes of operations; and wherein the software includes a virtual machine monitor for determining the mode of the I/O device prior to commencing devirtualization; and restoring the determined mode of the operation after the I/O device has been devirtualized.

65. The article of claim 56, wherein the software includes a virtual machine monitor for devirtualizing the I/O device at runtime.

66. The article of claim 65, wherein the virtual machine monitor can virtualize the I/O device after having devirtualized the I/O device at runtime.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.